ESD-TR-72-263

Semiannual Technical Summary

Speech

30 November 1972

Prepared for the Advanced Research Projects Agency
under Electronic Systems Division Contract F19628-73-C-0002 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

SPEECH

SEMIANNUAL TECHNICAL SUMMARY REPORT TO THE ADVANCED RESEARCH PROJECTS AGENCY

1 JUNE – 30 NOVEMBER 1972

ISSUED 11 JANUARY 1973

Approved for public release; distribution unlimited.

LEXINGTON MASSACHUSETTS

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Advanced Research Projects Agency of the Department of Defense under Air Force Contract F19628-73-C-0002 (ARPA Order 2006).

SUMMARY

A new formant tracking program is now available. Based on linear predictive analysis, it uses a formant enhancement technique to resolve merged formants. Time domain considerations are dealt with by working both forward and backward from the center of each high-energy voiced segment. The phonetics class segmentation program has been further developed and evaluated for a substantial set of utterances.

A program has been written for generating sentences appropriate to the system task for the Lincoln experimental speech understanding system. This task is the vocal command of the speech data retrieval, analysis and display system. A heuristic search program is being developed to support the linguistic analysis of sentences from this task domain. Input to the program is a noisy phonetic string representing the hypothesized output of a phonetic recognizer. A simple version of the program has successfully corrected a few test sentences.

Software for supporting the speech data base is operational. Programs for displaying the speech data and for manual labeling of the phonetic content are available. Work is under way on an automatic labeling procedure and a special network server for data hase users. The process of building up the data base contents has started.

The TSP system is now operating in a two-console configuration. The system is being used to log into TX-2 regularly and into other network hosts occasionally. Graphics output from the Lincoln IBM 360/67 has been displayed on a TSP console.

The TX-2 system has been extended to allow multiple network users. A new disk system to support the speech data base has been ordered, and work is under way on the necessary interfacing hardware and system changes.

Accepted for the Air Force
Joseph J. Whelan, USAF
Acting Chief, Lincoln Laboratory Liaison Office

CONTENTS

GLOSSARY

APEX	TX-2 time-sharing system
ARPA	Advanced Research Projects Agency
BCPL	Basic Combined Programming Language – intermediate-level language for computer programming
FDP	Fast Digital Processor – Lincoln Laboratory computer designed for waveform processing applications
LPC	Linear Predictive Coding – method for signal analysis being used in current speech research
SURNET	Specialized server process intended to provide network access to speech data base on TX-2
TELNET	Software which allows console on one network computer to function as console for another
TSP	Terminal Support Processor

SPEECH

I. PHONETIC RECOGNITION

Phonetic recognition work is proceeding on the Laboratory's FDP-1219 facility. The goal is to achieve a useful approximation to a phonemic transcription of the input speech waveform. During the current reporting period, work has concentrated on the development of programs for the phoneme class segmentations and formant tracking. Results have been encouraging and current work is being directed toward the more detailed study of individual phoneme classes (stops, nasals, etc.) with a view toward improving segmentation and proceeding toward identification. In terms of identification, particular attention is being paid to combining the segmentation results with formant results, in order to be able to measure formant slopes at segment boundaries.

The following sections present the formant tracking and segmentation algorithms in some detail and indicate the results of a preliminary test of the segmentation program.

A. Formant Tracking

A formant tracking algorithm, which yields the first three formant positions and amplitudes in voiced regions, has been implemented on the FDP-1219 facility. It uses as input the linear predictive coding (LPC) spectrogram, the LPC coefficients and the segmentation indicator. The algorithm has been tested on a large number of unrestricted sentences and rarely makes a mistake. It is occasionally unsuccessful in separating out two merged peaks, but in those cases it usually recognizes correctly which formant is missing. It gets confused sometimes in regions which were mistakenly labeled "voiced," as would be expected. Such confusion may be a strong suggestion that the segmentation indicator is wrong.

A LPC spectral frame is defined as the magnitude of the transfer function H(z) of the filter, represented by the LPC coefficients (a_i), evaluated at equally spaced points on the unit circle in the z-plane. That is, the transfer function

$$H(z) = \frac{a_o}{1 - \sum_{i=1}^{14} a_i z^{-i}}$$

is evaluated at $z = e^{j2\pi k(\Delta f/f_s)}$, where $k = 0, 1, \ldots, N-1$, $N = f_s/\Delta f = 256$, and $f_s = 10$ kHz. The resulting spectrum is a smooth curve with at most seven peaks, one for each of the seven pole pairs of the filter. There are almost always fewer than seven peaks since (1) some of the poles have a wide bandwidth and contribute only to frequency shaping, and (2) frequently two poles are close together and therefore merge into one peak.

Case (1) presents no problem, since wide bandwidth poles are not generally formants. However, it is necessary to recognize when case (2) has occurred, and in some way to verify that a given peak corresponds to two formants. In situations where it seems likely that two peaks have merged, an often successful approach to the problem has been to recompute the spectrum on a circle of radius $r < 1$ in the z-plane. This corresponds to evaluating H(z) at $z = re^{j2\pi k(\Delta f/f_s)}$, where $r < 1$. This serves to increase the amplitude of the individual peaks (enhance the formants) by moving in closer to the poles, and usually effeets a separation. This formant enhancement teehnique is analogous to, but simpler to implement than, the chirp-z transform method.[1]

The first step in the algorithm is to find all simple peaks in eaeh voieed spectral frame in the region from 150 to 3400 Hz. Eaeh frame may have one, two, three or four peaks. (If more than four peaks are present, only the first four are kept.) The task is to fill three formant slots with the appropriate peaks, keeping in mind the following:

(1) Sometimes two formants merge into one peak (common in non-vowel sonorants, such as r and w).

(2) Sometimes one formant is nonexistent (common in nasals).

(3) Sometimes spurious peaks are present (common in nasalized vowels).

(4) Many times the fourth formant is present to confuse the issue.

The approach is to have at hand an educated guess as to where the four formants are most likely to be in the frame under consideration, and to use this information to assign formant numbers to the available peaks, possibly leaving out some peaks and/or adding new peaks.

Since the algorithm relies heavily on the output from the previously proeessed frame to determine the output for the eurrent framc, it is neeessary to begin at a plaee where formants are most likely to be correet, and to move away from these toward the less eertain areas. Therefore, the program chooses the middle of each high cnergy voiced region as an anchor point, and branches out from there in both directions in time. Beginning with an initial educated guess consisting of fixed values for the four formant frequencies (F_1, F_2, F_3, F_4), formant slots are filled with available peaks on the basis of frequency position relative to the educated guess. After some special routines to check for unused peaks and/or unfilled slots, the educated guess is updated with the formant positions dccided in thc current frame, to be used in defining pcaks in the next frame. Formant slot S_4 is included because F_4 often falls below 3400 Hz. The slot is kept only to prevent eonfusion between F_3 and F_4, and no special attempt is made to fill S_4 if it is left unfilled after an initial pass. If any formant slot is left unfilled in the eurrent frame, then the frequeney for that formant is left unehanged in the edueated guess.

The following is a detailed step-by-step deseription of the algorithm:

Begin in the middle of the first high-volume voieed region, and set the "edueated guess" for F_i (i = 1 – 4) to initial eonditions: F_1 = 320 Hz, F_2 = 1300 Hz, F_3 = 2760 Hz, F_4 = 2300 Hz.

(1) Fetch input data:– Start with "educated guess" values for F_i (i = 1, 2, 3, 4) for the frequency positions of the first four formants, four empty slots S_i (i = 1, 2, 3, 4) for new formant positions, and the frequency positions and amplitudes of up to four peaks P_j found in the region from 150 to 3400 Hz in the current spectral frame.

(2) Fill slots:– Fill eaeh formant slot S_i with the best eandidate peak P_j by the following rule: The peak P_j closest in frequeney to F_i (i = 1, 2, 3, 4) goes into S_i. (In most eases, this will provide a 1:1 mapping of P_j into S_i, and steps (3) through (5) (below) will be unneeessary. However, steps (3) through (5) are often neeessary to resolve the typcs of ambiguities mentioned above.)

(3) Rcmove duplicates:– If the same peak P_j fills more than one slot S_i, keep it only in the slot S_k which corresponds to the formant F_k that it is elosest to in frequency, and remove it from any other slot(s).

(4) Deal with unassigned peaks:– If there are no unassigned peaks P_j, go to step (5). Otherwise, try to fill empty slots with peaks not assigned in step (2) as follows:

(a) If there is a peak $P_{j=k}$ unassigned, and a slot $S_{i=k}$ unfilled, fill the slot with the peak and go to step (5).

If there is a peak $P_{j=k}$ unassigned, but slot $S_{i=k}$ is already filled, check the amplitude of P_k as follows: If amplitude $(P_k) < 1/2$ amplitude (peak already assigned to S_k), throw P_k away and go to step (5). Otherwise, go to step (4b).

(b) If P_k is still unassigned, but $S_{i=k+1}$ is unfilled, move the peak in $S_{i=k}$ to $S_{i=k+1}$, and put P_k in S_k. Go to step (5).

(c) If P_k is still unassigned, but $S_{i=k-1}$ is unfilled, move the peak in $S_{i=k}$ to $S_{i=k-1}$, and put P_k in S_k. Go to step (5).

If steps (4a, b and c) all fail, throw P_k away.

(5) Deal with unfilled slots:– If S_1, S_2 and S_3 are all filled, go to step (6). (S_4 may or may not be filled.) Otherwise –

(a) Recompute the spectrum from the predictor coefficients with $r < 1$ to enhance the formants and hopefully separate two merged peaks. Go to step (1). Repeat steps (1) to (5) up to 6 times with radius = 0.98, 0.975, 0.97, 0.965, 0.96, 0.955. Usually, before 6 times, the empty slot gets filled and step (5) falls out to step (6).

If there is still an unfilled slot after the sixth attempt, reset the radius to 1, repeat steps (1) to (4) to restore the original formant positions and amplitudes, and go to step (5b).

(b) If the empty slot is S_3, and S_4 contains a peak and step (5a) has been tried 6 times, move the peak in S_4 to S_3 and go to step (6).

(6) Record answers:– Accept formant slot contents as answers for this frame. Also use formant slot contents as "educated guess" for next frame. (If a slot is empty, keep the original "educated guess" for that formant.)

(7) Fetch next frame:– Fetch the next spectral frame and go to step (1) unless

(a) Next frame is unvoiced.

(b) (Backward processing only) Next frame has already been processed by forward branch from previous anchor, or

(c) (Forward processing only) Next frame is the beginning of a new high volume voiced segment.

If step (7a, b or c) is true, go to step (8).

(8) Reset initial conditions and go to anchor:–

(a) If processing was backward, begin again at anchor and now process forward. Go to step (1).

(b) If processing was forward, find a new anchor in the middle of the next high volume voice region. Go to step (1).

After all the data have been processed, a second pass edits out obvious one, two or three frame errors in each formant, but only in areas where the four frames surrounding the unaligned or missing frames are relatively smooth.

The nonlinear smoothing method applied separately to each formant track is defined in detail as follows:

Let the frequency location of formant F_i in the n_{th} frame be L_n.

Define $D_{ab} = |L_{n+a} - L_{n+b}|$, a measure of the alignment of a particular formant.

Th = threshold = 240 Hz.

If $D_{n,n-1} <$ Th, frame n is considered smooth.

If $D_{n,n-1} >$ Th, an attempt is made to smooth frame n, but only if either (1), (2) or (3) is true:

(1) If $D_{n-1,n-2} <$ Th, $D_{n+1,n-1} <$ Th and $D_{n+2,n+1} <$ Th, replace L_n with $\frac{L_{n+1} + L_{n-1}}{2}$ and move to frame n + 1. (1 frame out of line)

(2) If $D_{n-1,n-2} <$ Th, $D_{n+2,n-1} <$ Th and $D_{n+3,n+2} <$ Th, replace L_n with $\frac{L_{n+2} + L_{n-1}}{2}$ and move to frame n + 1. (2 frames out of line)

(3) If $D_{n-1,n-2} <$ Th, $D_{n+3,n-1} <$ Th and $D_{n+4,n+3} <$ Th, replace L_n with $\frac{L_{n+3} + L_{n-1}}{2}$ and move to frame n + 1. (3 frames out of line)

The new L_n is used in evaluating frame n + 1.

Figures 1 and 2 illustrate the behavior of the formant tracker. Figure 1 shows the output of the formant tracker for the sentence "Rice is often served in brown (bowls)." The formants are written over the spectrogram and also displayed separately above it. Notice that formants are traced only during voiced sections of speech. Formant enhancement was needed in the /r/s of "rice" and "served," and in the /o/ of "often." Figure 2 shows a spectral cross section in the /r/ of "rice" before and after formant enhancement. Notice that F_3 appears as a distinct peak only after enhancement.

B. Segmentation

The segmentation program for separating speech into phonetic classes has been developed further, and has been evaluated for a substantial set of utterances.

The inputs to the segmentation algorithm are measurements on the speech spectrum and waveform, sampled every 6.4 msec to correspond with the times at which spectral cross sections are determined. A segmentation category is assigned to each spectral cross section, on the basis of suitable thresholding and combination of these measurements. Since the last report,[2] the segmentation algorithm has been refined somewhat, and the number of categories has been increased from four to nine.

The measurements now incorporated in the segmentation decisions are:

(1) $$v_T(n\Delta t) = \left[\frac{1}{128} \sum_{k=0}^{127} S_h^2 (k\Delta f,\ n\Delta t) \right]^{1/2}$$

= rms spectrum, 0 to 5000 Hz

(2) $$v_H(n\Delta t) = \left[\frac{1}{128} \sum_{k=8}^{127} S_h^2 (k\Delta f,\ n\Delta t) \right]^{1/2}$$

= rms spectrum, 320 to 5000 Hz

(3) $$r_1(n\Delta t) = \frac{\left[\frac{1}{128} \sum_{k=0}^{22} S_h^2 (k\Delta f,\ n\Delta t) \right]^{1/2}}{\left[\frac{1}{128} \sum_{k=85}^{128} S_h^2 (k\Delta f,\ n\Delta t) \right]^{1/2}}$$

= (rms spectrum, 0 to 880 Hz)/(rms spectrum, 3400 to 5000 Hz)

(4) $$sd(n\Delta t) = \frac{\left\{ \sum_{k=9}^{70} \left| S_h[k\Delta f,\ (n+1)\ \Delta t] - S_h[k\Delta f,\ (n-1)\ \Delta t] \right| \right\}}{\left\{ \sum_{k=9}^{70} S_h[k\Delta f,\ (n+1)\ \Delta t] + S_h[k\Delta f,\ (n-1)\ \Delta t] \right\}}$$

= spectral derivative, 360 to 2800 Hz

(5) $$vc = \begin{cases} 1 & \text{when pitch detector finds voicing} \\ 0 & \text{when pitch detector does not find voicing.} \end{cases}$$

In these definitions, $\Delta f = 10000/256 \approx 40$ Hz, $\Delta t = 6.4$ msec and $S_h(\varphi, \tau)$ is the amplitude at frequency φ of the homomorphic spectral cross section measured at time τ. The voiced-unvoiced decision vc is made via the Gold and Rabiner pitch detector algorithm.[3]

A segmentation category is assigned to each spectral cross section, on the basis of suitable thresholding and combination of these measurements. The segmentation categories, and a description of their meaning in terms of the measurements above, are as follows:

(1) VWL1 — V_H high, r_1 high, vc = 1

(2) ASP1 — V_H high, r_1 high, vc = 0

(3) DIPV — dip in v_T during VWL1 segment

(4) DIPU — dip in v_T during ASP1 segment

(5) VFRC — r_1 low, vc = 1

(6) FRC — r_1 low, vc = 0

(7) VBAR — v_H very low, vc = 1

(8) SIL — v_H very low, vc = 0

(9) SPDR — boundary marker indicating a sharp peak in sd during WVL1 or DIPV segment.

Some editing of these segment indicators is done to eliminate spurious results. For example, any category which persists for only one look is eliminated and set equal to an adjacent category. Also, some simple time smoothing of the v_T, v_H and sd measurements is done before thresholding.

The performance of the segmentation program was analyzed in detail for a set of 25 sentences spoken by a single male talker. The sentences were chosen to have representative samples of vowels, fricatives, stops and nasals. The results of this analysis, organized by phonetic classes, will now be discussed. Results for other speakers and other sentences have been obtained and are similar, although such things as threshold levels seem somewhat speaker-dependent.

Essentially, all the vowels in the sentences were marked as VWL1. Observations of vowel detection on other speakers indicate that comparable results can be obtained if some adjustment of the frequency ranges in the numerator and denominator of r_1, and some changes in threshold levels, are allowed.

The sentences contained 33 strong fricatives (/s/, /sh/, /z/, /zh/) which were all marked as FRC or VFRC. Similar comments hold on extension to other speakers. The results on the weak fricatives (/f/, /<u>th</u>in/, /v/, /<u>th</u>en/) were quite inconsistent. These were often marked as SIL, VBAR, DIPV or DIPU, as well as FRC and VFRC. This behavior reflects the fundamental difficulty that the weak fricatives have little energy and are not articulated clearly.

About 80 stop consonants were contained in the sentences, of which about 60 were detected by either a SIL or VBAR indicator. These indicators (SIL and VBAR) gave 5 false alarms. Particular difficulty was encountered in detecting intervocalic flapped /t/s, and stops adjacent to nasals, since in these cases the stop closure was often so short that V_H did not drop below the threshold. Different measurements will be needed to detect these phenomena.

Of the 36 nasals in the sentences, 28 were marked either as DIPV or by SPDR boundaries. However, additional work is being carried on to identify the nasals as a distinct phonetic class, since other sounds (such as voiced fricatives and glides) are often marked by these same indicators.

II. LINGUISTICS

The goal of our work in the linguistics area is to explore the value of restrictions on vocabulary, syntactic freedom, and semantic complexity in interpreting the output of a noisy phonetic recognizer. One plan is to carry out a series of experiments to gain quantitative data about the information available for correcting errors and resolving ambiguities at the various possible levels of linguistic processing. These experiments will be carried out in a linguistic environment constrained by the task we have set for our experimental speech understanding system. This task is the vocal command of our speech data retrieval, analysis and display system. We believe that this task environment is well suited to our experimental requirements, since it can be varied considerably in vocabulary size, syntactic freedom and semantic complexity, and still retain useful properties.

To support the experiments, we are developing a number of program modules, some of which may serve as components of our speech understanding system. Others will be used only in the experimental work. These programs will allow us to (a) generate sentences appropriate to our task domain, (b) garble these sentences according to the hypothesized output of a phonetic recognizer and (c) process these garbled phonetic strings at lexical, syntactic and semantic levels to reconstruct the sentences.

Work on sentence generation is intended both to provide phonetically transcribed sentences in sufficient quantity for statistical studies and to gain insight into the complexity of the semantic model required to produce reasonable sentences within our restricted linguistic environment.

The work on garbling is an attempt to simulate not one particular, but a set of hypothesized phonetic recognizers. The intent is to assess the relative sensitivity of higher level processing to the accuracy and detailed error characteristics of the input data.

Work in lexical segmentation and parsing will attempt to separate the various levels of analysis during this experimental work so as to increase our understanding of the mechanisms at work in linguistic processing, even though we feel that a more unified approach is likely to be more efficient in an actual system.

To date, some progress has been achieved in sentence generation and lexical segmentation. The following two sections will discuss those activities in greater detail. Our work in the parsing area is just beginning with the implementation on TX-2 in BCPL of R. Kaplan's General Syntactic Processor (GSP) system.[4] We feel that the GSP system will be a useful tool for supporting our experimental work, since it can be used to emulate a variety of parsing schemes.

A. Sentence Generation

A program has been written for the automatic generation of sentences appropriate to the task we have set for our experimental speech understanding system. The sentences are all commands to the system for data retrieval, calculations and manipulation of data, or for output and presentation of data. The system being commanded consists of a retrieval program which can search on sample number, speaker number or speaker name, utterance spoken, specific phoneme string, or a combination of these. In addition, programs can call for the calculation (if needed), display and printing of such things as waveform, spectrogram, pitch, formants, zero crossing, first movement, amplitude, etc. Also, the system can make a catalog of any sections of data of special or current interest. The display system can exhibit any markers and/or labels it has, and can modify, delete or add to its display information. A cursor is used to identify a location of current interest. Calculations and modifications can be temporary or can be made permanent items in the data base.

The current vocabulary being used to make up English language commands to this system consists of 50 verbs, 51 nouns, 9 prepositions, 13 adjectives, 7 adverbs, 3 determiners and 29 number words. Each word in the vocabulary is stored with its lexical and phonemic transcription, part of speech, subject matter (as, "utterance," "display," etc.), syntactic features (as, "takes object") and special cases (as, "'from' needs a 'to';" "'speaker number' requires a number"). The program first picks a verb at random. As soon as the verb is picked, only certain nouns can be the subject or object, and only some prepositions can form modifying phrases. For example, if the verb is one for bringing in data, nouns such as "read," "utterance" or "sentence number" would be possible objects, but "scope" or "fricative" would not.

As yet, the program does not form sentences with a subject other than the implied "you" of commands. Verbs which require subjects are bypassed. Thus, if the verb takes an object and/or an indirect object, the program picks at random from the possible nouns for the object(s). If the verb can take a modifier, the program decides (chances 1 to 1) whether to use an adverb, and if so, chooses at random from the possible adverbs. Some verbs are marked "sentence complete with object," as "refresh" (the graph) and "clear" (the scope). If it is not so marked, a preposition is chosen at random from those permitted. Restrictions will allow "read to" or "read the sample with," but not "read of" or "read the sample on." These groups of permitted words have been called classes, for lack of a better word. Each preposition sets a new class of words as possible objects, modifiers, and next preposition. Any noun has a 1-in-3 chance of having a modifier, unless it is a data retrieval sentence. In such cases, because of the need for more modifiers to specify what is desired, all nouns have modifiers, if possible. The determiner is "the" unless the verb or preposition specifies "either 'the' or 'a,'" or "no article." Nouns are coded "use 'an' not 'a.'" Some special cases exist. One is that "speaker number" and "sentence number" do not take beginning modifiers, but are followed by a random number between 1 and 999. Another is that "change" and "from" require a "to" to appear later in the sentence. "Sequence" and "string" are either followed or preceded by 2 or 3 phoneme names. The latter case is also the indication of the end of the sentence. The end will occur after three prepositional phrases or when a phoneme name is the object of the preposition.

To follow through on the generation of a sentence – a verb is picked at random from all possible verbs. Suppose "mark" were picked. This verb would be coded in the dictionary as "to be followed by 'this a'" and "takes an object of the phoneme class." The object would then be picked at random from the phoneme names listed. Suppose "aspirate" were picked. This name would be marked "use 'an' not 'a.'" So, the sentence thus far would be: "Mark this an aspirate." The program would then look for possible prepositions. Two would be marked as possible following a phoneme name: "from" and "to." Each of these would be marked as taking a phoneme name as their object, and "from" would be marked "must have 'to' following object." One of these objects, and then a phoneme name would be picked at random. So the sentence may end as, "Mark this an aspirate to the vowel," or "Mark this an aspirate from the silence to the nasal."

If the initial verb had been "retrieve," the verb would have been marked "takes 'the' or 'a' as determiner." The system would then decide at random which to use. Possible objects would be nouns such as "data," "example," "phrase," "sample," "sample number," "sentence," "statement," etc. Suppose "sample" were picked. This object could be followed by "containing" (perhaps followed by a phoneme name sequence), "by" (and a description of the speaker), "with" (and again perhaps a phoneme string of some kind), etc.

Some examples of acceptable sentences are:

> "Call this section vowel from the silence to the liquid."
>
> "Print the spectrogram of the speaker with the aspirate-vowel sequence."
>
> "Change the transition to a voicing."
>
> "Get me the utterance with the liquid-voicing-plosive segment."

Some sentences which indicate remaining problems are:

> "Xerox the amplitude on the graph of the envelope for the vowel."
>
> "Put up the distribution for the fricative to the voicing."
>
> "Catalogue the string of fricatives from the liquid to the plosive."

B. Lexical Segmentation

The problem of transforming a continuous string of phonemes (i.e., without word boundaries) into a set of sequences of words (to be analyzed by the syntactic and semantic components) is being investigated from the viewpoint of heuristic search.[5]

1. Heuristic Search

The casting of the problem in the heuristic search framework is done by considering each partial segmentation (i.e., some, but not all, word boundary decisions have been made) as a state or node. The original string is the starting state and a new state is created when a new decision is made about the placement of a word boundary. There is always a current state from which new states are generated. Normally, several sub-states are generated, and we have a tree, the terminal nodes of which represent alternate decisions about word boundaries.

Each node is assigned a value by an evaluation function and the node with the highest value is made the current node. This is repeated until a node is generated that satisfies an acceptance criterion.

The use of this approach allows the more promising possibilities to be examined first, and with it we hope to overcome the combinatorial explosion problem often encountered in search strategies.

2. Confidence Numbers

The heuristic search approach provides us with a framework in which we can perform the segmentation in some order other than left-to-right, but the choice of order would have to be random unless we get added information from the front-end phoneme recognizer.

Fortunately, we believe that it is feasible for the front end to provide us not only with its determination of a phoneme, but also with a confidence number, which is an estimation of the reliability of this determination.

Using these confidence numbers, we can select as the initial parts of the string to be examined those regions which seem most likely to be correct.

3. Centering

Since we have no idea where in a given string a word begins, we propose to use the technique of choosing the phoneme that has the highest confidence number as an anchor point, finding all words in the dictionary that contain this phoneme, and matching each word against the string by aligning the phoneme common to the two sequences.

4. Matching

The match program must take into account such factors as: possible insertions and deletions, the likelihood of a word boundary given prosodic or phonological information, and the likelihood of confusion if the phoneme in the word does not match the phoneme in the input.

5. Generative Recognition

Since future continuations are strongly constrained by previous selections, we wish to be as certain as possible of the first few nodes in a branch. To enhance the effectiveness of our evaluation, we propose to pass words that are given high scores by the match program to a generative recognition program that will compare the word (using information stored in the dictionary) with some representation of the original acoustic input. This analysis-by-synthesis approach should enable us to narrow the number of words that are finally selected to a very small number.

The generative recognition program will be an expensive procedure (in terms of time), but we expect to execute it mainly in the first few levels where the confidence numbers in the regions examined are high.

6. Parsing

The output of the lcxical segmentation phase is not a sentence, but a sequence of sets of words. This will result from our decision to keep not just the highest scoring word in a region, but all those meeting our criteria. We expect that this factored form of a class of possible sentences is a more useful representation than a list of sentences, since the parser may now be able to work on sets of sentences rather than one at a time.

7. Current Implementation Status

A garbler has been written that will take scntences in lexical form, look up the phonemic transcription and randomly garble a given percentage of phonemes. A heuristic search program has bcen implemented that uses a very simple evaluation function, but has been able to dcgarble some sentences in our domain. Currently, no phoneme insertions or deletions are hypothesized by cither program.

Work is proceeding on allowing insertions and deletions, better use of generated confidence numbers, and a more sophisticated matching and evaluation program (including generative recognition).

III. SPEECH DATA BASE

The Speech Data Base is intended to provide fast, automatic access to the entire range of data associated with each of many utterances. The supporting software is now operational and the data base is available to users of TX-2 both locally and from the ARPA network via TELNET connection. Recent work has been concerned with programs for reading speech into the data base, for displaying the results of processing, and for manual labeling of the phonetic contents of utterances. Work is under way on development of SURNET, a special network server to provide more efficient remote access to the data base and a scheme for automating the labeling process. These activities and the plans for building up the data base contents are discussed in the following sections.

A. Data Base lnput

Programs for reading speech into the data base and for writing data to tape have been written and checked out. Spcech on analog tape is converted into a digitized waveform via the TX-2 A/D converter and stored in the data base. The waveforms are then retrieved and written to tape for archival storage and/or FDP processing. The FDP produces an output tape containing the original waveform and the results of processing (linear predictive and homomorphic spcctra, pitch, formants and segmentation). These results are read into the data base and stored as fields associated with the original waveform.

The analog converter samples at 10 and 20 kHz to produce two waveforms for each utterance. Two different sampling rates are required to meet the needs of thc Lincoln work, as well as that of other projects having network acccss to the data.

B. Speech Data Display Facility

The Speech Data Display Facility is essentially complete, although some changes will comc about with use. Since it runs under the spcech processing controller and interfaccs with the data basc, it can work together with analysis, search and other facilities. lt uses both the Tektronix 611 and Hughes scan converter displays and can be driven via tablet or keyboard.

Time synchronized spectrograms, spectral cross sections, and functions of time can be displayed. X and Y scales on the latter two types of graphs can be adjusted by the user.

The screen can be dynamically compartmentized into any number of rows and columns, enabling the user to get as many graphs as possible onto the screen face. This technique reduces the frequency of repainting, an important consideration when dealing with storage scopes.

Each graph is completely identified. A heading gives the utterance and data type, as well as certain display parameters. The time axis is displayed with tick marks and labels at tenth or hundredth of a second intervals. The frequency domain is marked off at 1-kHz intervals.

Eight types of commands exist for selecting utterance and data type, setting display parameters, editing the screen face, etc. There are facilities for manually labeling, or editing an existing label field. A flexible system of global display variables and defaults enables the user to get things done with a minimum of explicit commands.

C. Spectrograms on a Two-Level Storage Scope

The wide availability of the Tektronix 611 storage scope has motivated an effort to produce a usable spectrographic display with this medium. By using six levels of brightness (realized by a half-tone technique) and enhancing spectral peaks as a normalization procedure, we have obtained displays which show promise for identifying phonetic events. By using the LPC based spectra, we assume a controlled number of peaks in the spectrum (6 for the data we are using). The resulting displays do not mimic the traditional analog spectrograms, but have many of the same general properties. A notable difference lies in the representation of the voiceless consonants.

Figure 3 shows a composite of four storage display spectrograms obtained with different parameter settings. The pictures are made by the Tektronix 4601 hard copy unit connected to the 611 display scope. Two scope widths are required to encompass an entire sentence of this length (1.5 sec). The amplitude function and phonetic labeling are shown at the bottom of the figure.

D. Speech Labeling Facility

From the beginning, the Speech Labeling Facility has been considered a cornerstone of the Data Base. It consists of identifying the phonemes and words in an utterance and specifying their start and end times. Thus, this function lies precisely at the crucial interface between acoustic data and linguistic entities.

A graphic display facility has been built to enable manual labeling and editing. But this cannot be sufficient with respect to speed, accuracy and consistency, when one considers the quantity of speech data we expect to put into the data base and the time available to do manual labeling. Therefore, an automatic labeling facility has been designed.

Input to the labeling program will consist of:

(1) The lexical transcription of the utterance in the form of a lexical array.

(2) A dictionary containing the phonemic representation of the words.

(3) A "nucloid" time event array, which presents the results of gross acoustic processing: syllable nuclei, unvoiced stretches, voiced stretches and silence.

The program will generate a phonemic label array and then match this up with the nucloid array, assigning start and end times to each segment. Tentative matches will be evaluated by gcncrating estimated times for the phonemic segments and seeing how well these times fit the acoustic data. Finally, start and end times will be inserted in the lexical label array.

Some words have different pronunciations according to the number of syllables used; e.g., "library" may be pronounced using two or three syllables. And, of course, function words may be "swallowed," i.e., lose syllabicity. To approach this problem, a tree-structured organization will be used to allow for the examination of alternative segmentations.

E. SURNET

Mention was made of a specialized network server facility called SURNET in an earlier report.[2] The SURNET server is intended to provide multiple user network access to the data base without the necessity of logging into the TX-2 as a normal (TELNET) user. At the time of the last report,[2] a SURNET design plan had been prepared for presentation at the June 1972 meeting of the ARPA Speech Data Base Working Group. Feedback from the Working Group and the network community has led to a modification of the original design. The current design, which is now being implemented, is summarized here.

The server consists of two parts. One part acts as a receptionist waiting for incoming calls to establish network connections. On receiving a call, the receptionist program, following standard initial connection protocol, attempts to establish a duplex connection between the user proccss and the SURNET server. Once established, the two-way connection provides the user process with a transmission path for sending commands to SURNET and receiving responses back.

The sccond part of SURNET interprets and executes the commands sent over the established connections. Two basic commands will bc implemented initially and others will be added as needed. The two basic commands are "query" and "store." The first provides thc user process a means for querying the speech data base (SDB). The second allows a user process to store data into the data base. All transmissions will follow the standard network data transfer protocol and individual data items will be converted (if required) to agree with Data Base Working Group conventions.

F. Data Base Content

Currently, the data base contains only a few tcst utterances which are being used for program dcbugging purposes. The plan is to build up toward a useful quantity (≈200) of labeled sentences by spring 1973. A plan for selecting a reprcsentative sample of speech of interest to the overall ARPA speech community has been worked out by the Data Base Working Group. Each participating organization has recorded a set of fifty sentenees relating to the task domain of interest to that organization. A sub-group of the Working Group is in the process of selecting a subset of these sentences which contain an appropriate distribution of phonetic and linguistic properties. This subset will serve as the nucleus of the common portion of the data base. Other speech may be entered by individual organizations for their own use. In particular, we expect to cnter speech related to the Lincoln task domain discussed in Sec. II-A, as well as some artificially contrived sentences intended to contain particular combinations of phonetic elements of interest in our phonetic recognition research.

IV. SYSTEM ACTIVITIES

A. TSP System

The TSP system is now operating in a two-console configuration. Operational software includes a process scheduler, a keyboard handler, a storage scope text output handler, a user command interpreter, a network control program and a user TELNET. The system is being used to log into TX-2 regularly and into other network hosts occasionally. A program to translate graphic output from the Lincoln IBM 360/67 in Stromberg-Datagraphics 4060 plotter format has been demonstrated. Work is starting on the TSP disk handling routines.

B. ARPA Network

1. TELNET Facilities

The TELNET SERVER and LOGGER programs, which allow ARPA Network log in to TX-2, have been reorganized and expanded to permit more than one remote user to access TX-2 at the same time. The reorganization involves a dispatcher subprogram which receives the calls from the network user and dispatches to one of the ultimate servers. Currently, two network users may simultaneously log in and run programs on TX-2. The new program organization makes it a simple matter to expand the number to three or four should the need arise.

The dispatcher subprogram will also receive calls and perform the appropriate dispatching for the SURNET speech data-base service and other services which may be provided in the future.

Thus far, the TX-2 SERVER facility has provided a convenient vehicle for debugging the TSP system, as well as for serving a regular remote user.

A TELNET USER program which enables one to log in to remote network hosts has reached full operational status. A variety of operating options provide for control of input characteristics (choice of erase character, input from text-file and others), of typewriter echoing, and of destination for responses from the remote host (typewriter, scope, printer, text-file, and others).

2. Data Rate Measurement

We have performed a study to determine the efficiency of the TX-2 Network Control Program (NCP) in the area of bit throughput rate. We wrote a matched pair of simple programs, a sender and receiver, designed to run on TX-2 and communicate with each other as fast as possible using standard host-to-host protocol. These programs were run for different combinations of TX-2 virtual machines to explore the effects of overlapped requests for service. The results are shown in Fig. 4. These results show that the data handling capacity of the TX-2 NCP will not be a limiting factor in exchanging data with other network hosts.

C. TX-2 System Changes

The storage requirements of the speech data base will be met by a single spindle IBM 3830/3330 disk memory system. This unit, which is scheduled for delivery in the second quarter of next year, will be connected to a new TX-2 I/O channel adapter which will realize most of the functions of an IBM block multiplexer channel with eight sub-channels.

The TX-2 cycle stealing I/O processor has been moved to a separate memory bus processor port in order to reduce memory access conflicts with the TX-2 CPU. The I/O processor is being further modified so that many of the increased programming abilities of the block multiplexer sub-channels can be available in all the I/O cycle stealing channels.

D. BCPL

Work on BCPL, the principal language bcing used for TX-2 speech software, has resulted in improvements to both the language and the compiler. Most notable to users has been a reorganization of the code generator part of the compiler that reduced total compilation time by about 40 percent. The quality of the code cmitted by the compiler has also improved considerably. It occupies less space and runs faster, in a few cases dramatically so.

REFERENCES

1. R.W. Schafer and L. R. Rabiner, "System for Automatic Formant Analysis of Voiced Speech," J. Acoust. Soc. Am. 47, 634 (1970).
2. Speech Semiannual Technical Summary, Lincoln Laboratory, M.I.T. (31 May 1972), DDC AD-745970.
3. B. Gold and L. R. Rabiner, "Parallel Processing Techniques for Estimating Pitch Periods of Speech in the Time Domain," J. Acoust. Soc. Am. 46, 442 (1969).
4. R. Kaplan, "A General Syntactic Processor," to appear in National Language Processing, R. Rustin, cd. (Prentice-Hall, Inc.), in press.
5. A. Newell and G. Ernst, "The Search for Generality," in Information Processing 1965: Proceedings of IFIP Congress 1965, W.A. Kalenick, ed. (Spartan Books, 1965), Vol. 1, pp.17-24.

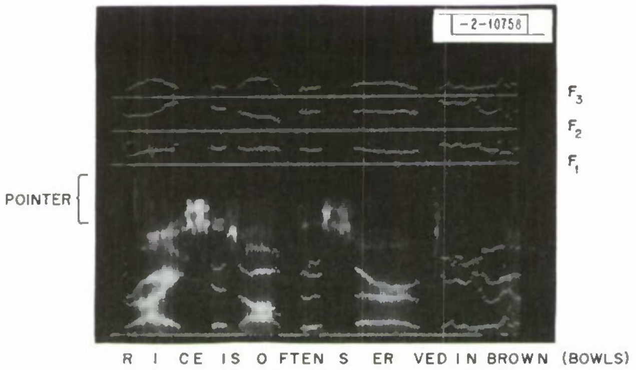
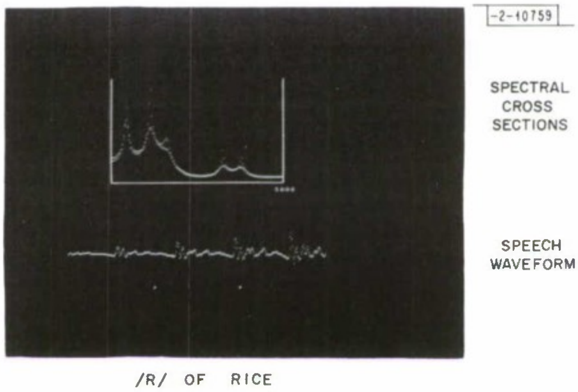


Fig. 1. Output of formant tracker shown superimposed on an LPC spectrogram and as separate functions of time.

Fig. 2. Spectral cross sections before and after formant enhancement and speech waveform for time slice indicated by pointer in Fig. 1.



I-2-10760

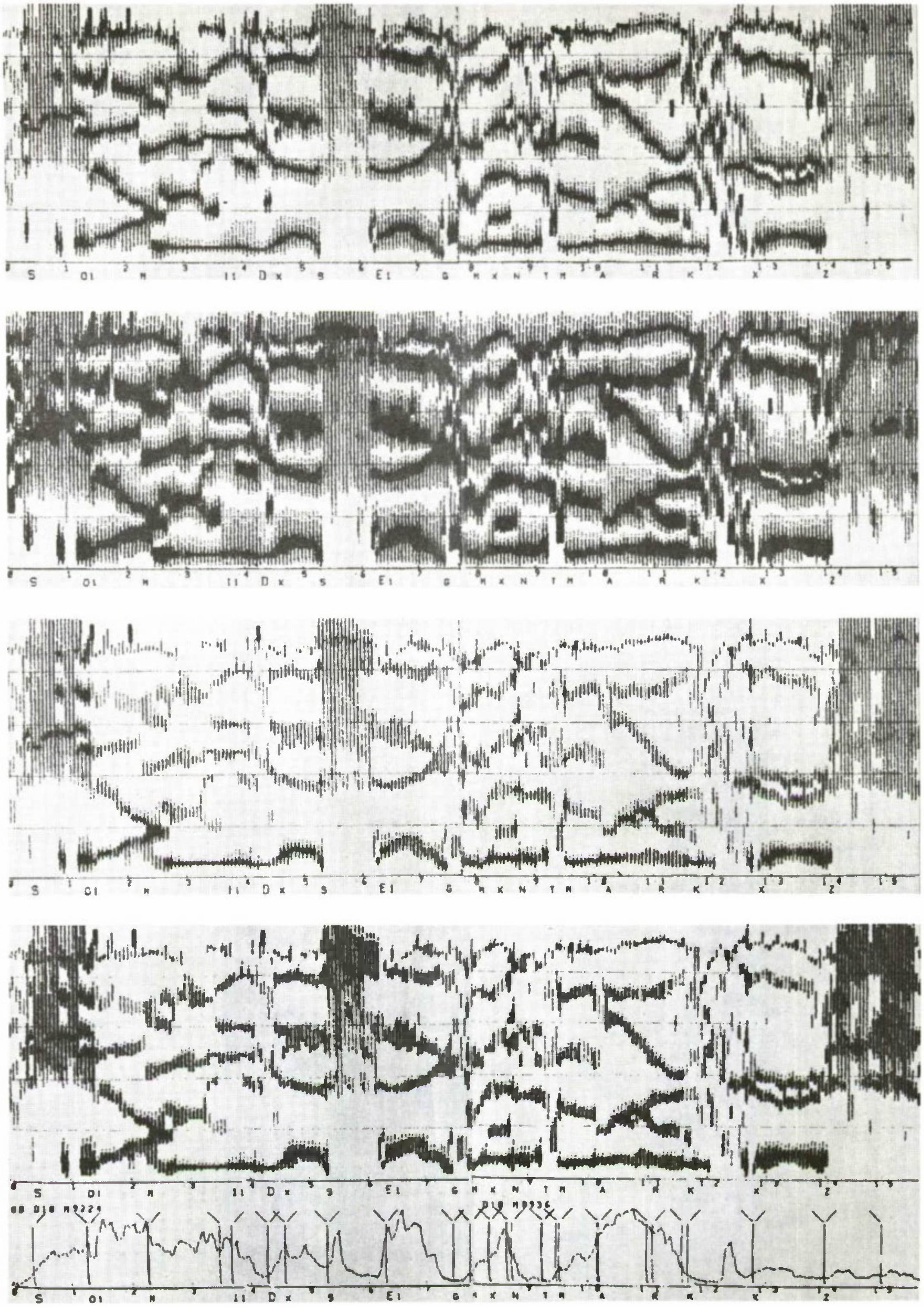


Fig. 3. Expcrimental spectrograms of sentence, "Show me the segment markers," made on Tektronix 4601 hard copy unit connected to Tcktronix 611 storage display scope. Four combinations of gain and normalization illustrate range of possibilities being explored. Overall amplitude and phonetic labeling are shown at bottom.

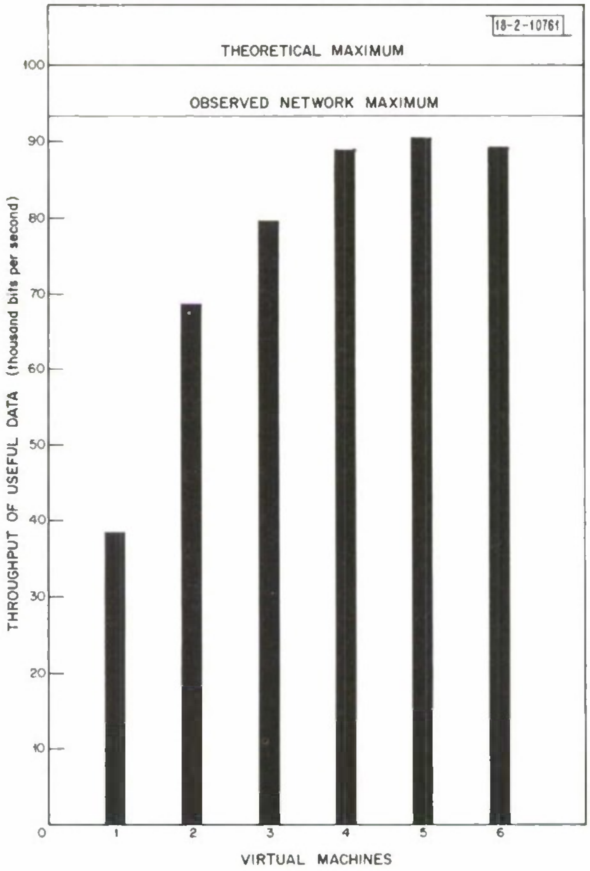


Fig. 4. Measured data handling capacity of TX-2 Network Control Program. Value of Observed Network Maximum is determined by current hardware capabilities and protocol overhead.

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Lincoln Laboratory, M.I.T.	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP None

3. REPORT TITLE

Speech

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Semiannual Technical Summary, 1 June through 30 November 1972

5. AUTHOR(S) *(Last name, first name, initial)*

Forgie, James W.

6. REPORT DATE 30 November 1972	7a. TOTAL NO. OF PAGES 24	7b. NO. OF REFS 5

8a. CONTRACT OR GRANT NO. F19628-73-C-0002 b. PROJECT NO. ARPA Order 2006 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Semiannual Technical Summary 30 November 1972 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* ESD-TR-72-263

10. AVAILABILITY/LIMITATION NOTICES

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency, Department of Defense

13. ABSTRACT

A new formant tracking program is now available. Based on linear predictive analysis, it uses a formant enhancement technique to resolve merged formants. Time domain considerations are dealt with by working both forward and backward from the center of each high-energy voiced segment. The phonetics class segmentation program has been further developed and evaluated for a substantial set of utterances.

A program has been written for generating sentences appropriate to the system task for the Lincoln experimental speech understanding system. This task is the vocal command of the speech data retrieval, analysis and display system. A heuristic search program is being developed to support the linguistic analysis of sentences from this task domain. Input to the program is a noisy phonetic string representing the hypothesized output of a phonetic recognizer. A simple version of the program has successfully corrected a few test sentences.

Software for supporting the speech data base is operational. Programs for displaying the speech data and for manual labeling of the phonetic content are available. Work is under way on an automatic labeling procedure and a special network server for data base users. The process of building up the data base contents has started.

The TSP system is now operating in a two-console configuration. The system is being used to log into TX-2 regularly and into other network hosts occasionally. Graphics output from the Lincoln IBM 360/67 has been displayed on a TSP console.

The TX-2 system has been extended to allow multiple network users. A new disk system to support the speech data base has been ordered, and work is under way on the necessary interfacing hardware and system changes.

14. KEY WORDS

speech understanding system
linear predictive coding (LPC)
SURNET
TELNET